

PETUNJUK PRAKTIKUM BASIS DATA

PP/TIF/BD/III/R2



Disusun oleh :
Dewi Soyusiawaty
Andri Pranolo

Laboratorium Basis Data
Program Studi Teknik Informatika
Fakultas Teknologi Industri
UNIVERSITAS AHMAD DAHLAN
2018

Kata Pengantar

Puji dan syukur kita panjatkan kepada Allah SWT yang telah memberi rahmat dan hidayahNya sehingga penyusunan revisi petunjuk praktikum Basis Data ini akhirnya bisa diselesaikan. Modul ini disusun sebagai panduan untuk pelaksanaan praktikum mata kuliah Basis Data di lingkungan Program Studi Teknik Informatika Universitas Ahmad Dahlan.

Materi yang disajikan sudah diurutkan disesuaikan dengan perencanaan mata kuliah tersebut, sehingga insyaAllah mahasiswa dapat dengan mudah memahami. Pada setiap pertemuan diberikan penjelasan tentang teori terkait materi yang diberikan dan langkah praktikum berisi tahapan kegiatan yang harus dilakukan mahasiswa/praktikan pada saat praktikum.

Penulis menyadari masih banyak ketidaksempurnaan pada penulisan ini, baik isi maupun redaksinya, oleh karenanya kritik dan saran yang membangun diharapkan dapat memperbaiki untuk tahun-tahun berikutnya.

Terima kasih kepada semua pihak yang telah membantu baik secara langsung ataupun tidak terhadap terselesaikannya petunjuk praktikum ini. Akhir kata, insyaAllah dapat bermanfaat bagi siapa saja yang membutuhkannya.

Penyusun

DAFTAR ISI

Kata Pengantar.....	i
Daftar Isi.....	ii
Pemahaman Kasus dengan Identifikasi Entitas.....	1
Pengolahan Data	8
Relasi Tabel.....	15
PHP & MySQL	19
DDL dan DML.....	31
Fungsi Agregasi.....	41
Query dari Relasi Tabel.....	45
Relasi Tabel dengan Join.....	50
Subquery.....	54

Pemahaman Kasus dengan Identifikasi Entitas

Pertemuan ke : 1

Alokasi Waktu : 2,5 jam (150 menit)

Kompetensi Dasar :

1. Praktikan mampu mengenal MySQL dan *Database Management System*.
2. Praktikan mampu memahami tabel, tipe data dan primary key dalam *Database Management System*.
3. Praktikan mampu mengidentifikasi entitas pada suatu kasus yang disediakan.

Indikator : Create Database, Create Table

A. Pendahuluan

1. Tabel dan Atribut

Entitas atau tabel merupakan individu atau objek yang memiliki sesuatu yang nyata (eksistensinya) dan dapat dibedakan dari sesuatu yang lain. Atribut adalah karakteristik atau ciri yang mendefinisikan entitas sehingga dapat membedakan antara entitas satu dengan entitas lainnya. Dalam sebuah entitas pasti memiliki satu atau lebih atribut untuk mendefinisikan karakteristik dari entitas tersebut. Contoh atribut dari Entitas Mahasiswa yaitu nama, nim, kelas.

2. Mengetahui MySQL

MySQL adalah Sebuah program database server yang mampu menerima dan mengirimkan datanya sangat cepat, multi user serta menggunakan perintah dasar SQL (Structured Query Language). MySQL merupakan dua bentuk lisensi, yaitu Free Software dan Shareware. MySQL yang biasa kita gunakan adalah MySQL Free Software yang berada dibawah Lisensi GNU/GPL (General Public License). MySQL merupakan sebuah database server yang

free, artinya kita bebas menggunakan database ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensinya.

MySQL pertama kali dirintis oleh seorang programmer database bernama Michael Widenius. Selain database server, MySQL juga merupakan program yang dapat mengakses suatu database MySQL yang berposisi sebagai Server, yang berarti program kita berposisi sebagai Client. Jadi MySQL adalah sebuah database yang dapat digunakan sebagai Client maupun server. Database MySQL merupakan suatu perangkat lunak database yang berbentuk database relasional atau disebut Relational Database Management System (RDBMS) yang menggunakan suatu bahasa permintaan yang bernama SQL (Structured Query Language).

Database MySQL memiliki beberapa kelebihan dibanding database lain, antara lain :

- a. MySQL merupakan Database Management System (DBMS).
- b. MySQL sebagai Relation Database Management System (RDBMS) atau disebut dengan database Relational.
- c. MySQL Merupakan sebuah database server yang free, artinya kita bebas menggunakan database ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensinya.
- d. MySQL merupakan sebuah database client.
- e. MySQL mampu menerima query yang bertupuk dalam satu permintaan atau Multi Threading.
- f. MySQL merupakan Database yang mampu menyimpan data berkapasitas sangat besar hingga berukuran GigaByte sekalipun.
- g. MySQL didukung oleh driver ODBC, artinya database MySQL dapat diakses menggunakan aplikasi apa saja termasuk berupa visual seperti Visual Basic, PHP dan Delphi.

- h. MySQL merupakan Database Server yang multi user, artinya database ini tidak hanya digunakan oleh satu pihak orang akan tetapi dapat digunakan oleh banyak pengguna.
- i. MySQL mendukung field yang dijadikan sebagai kunci primer dan kunci unik (Unique).
- j. MySQL memiliki kecepatan dalam pembuatan table maupun update table.

3. Pemahaman Kasus pada Sistem Informasi Perpustakaan

Saat meminjam buku di perpustakaan, seseorang akan melakukan registrasi terlebih dahulu. Kemudian, data peminjam akan disimpan oleh sistem dalam basis data. Setelah itu baru dapat dilakukan berbagai transaksi yaitu peminjaman buku, perpanjangan peminjaman buku, perpanjangan peminjaman buku, pengembalian buku, dan pembayaran denda. Seluruh transaksi yang dilakukan disimpan dalam basis data dan dapat diakses kembali oleh sistem sesuai dengan kebutuhan.

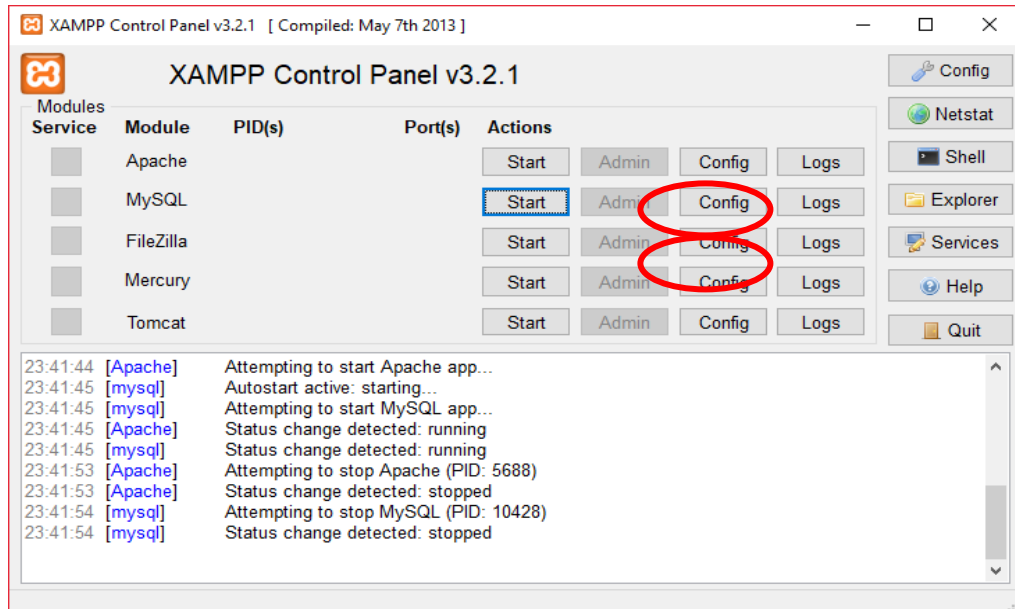
Peminjaman dapat mencari buku yang akan dipinjam dengan menekan tombol find atau search. Setelah itu sistem informasi akan berinteraksi dengan basis data perpustakaan untuk mencari buku tersebut. Lalu hasil pencarian akan ditampilkan pada layar. Melalui interaksi yang terjadi antara sistem informasi dengan basis data, petugas perpustakaan dapat mengetahui secara otomatis jika peminjaman buku lebih dari ketentuan yang berlaku, peminjam belum mengembalikan buku tetapi akan meminjam lagi buku yang lain, dan pemberitahuan ada yang kena denda pada hari H. Data yang disimpan dalam basis data meliputi :

Data buku, misalnya kode buku, judul buku, pengarang, penerbit, dan tahun terbit.	Data transaksi perpanjangan peminjaman, misal tgl perpanjangan, kode buku, jml buku diperpanjang, no peminjam, perpanjangan ke berapa, dan alasan
Data peminjam, misalnya nomor peminjam, nama peminjam, alamat, nomor telepon, dan email	Data transaksi pengembalian buku, misal tgl pengembalian, kode buku, jumlah buku yang dikembalikan, nomor peminjam.
Data petugas perpustakaan, misalnya nomor induk karyawan, jabatan dan level.	Data transaksi pemb denda, misal tgl denda, kode buku, jml buku yang kena denda, nomor peminjam, status lunas.
Data transaksi peminjaman, misalnya tanggal pinjam, kode buku, jumlah buku, nomor peminjam, dan lama pinjaman.	

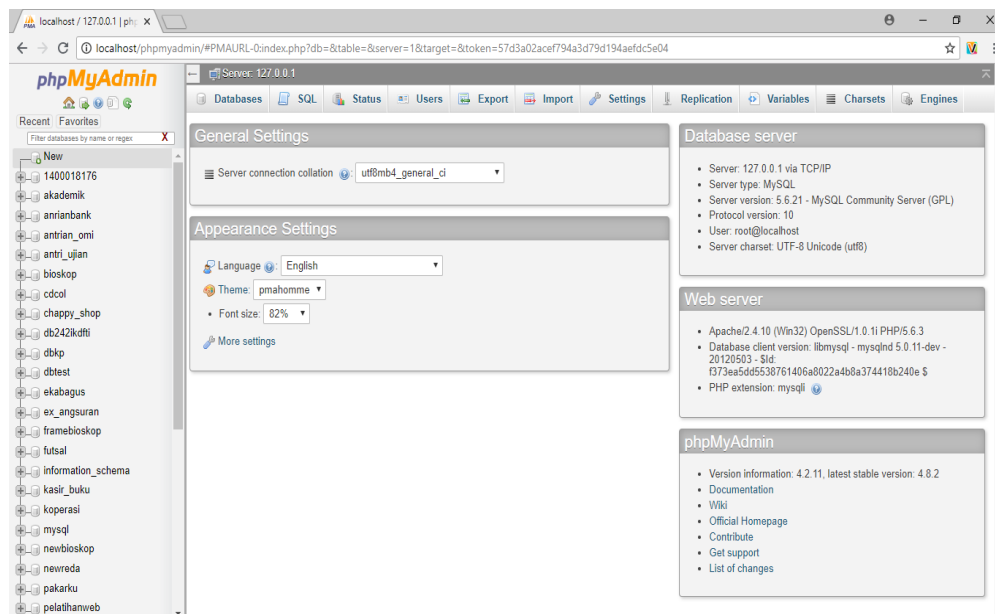
B. Petunjuk Praktikum

1. Mengakses PHPMyAdmin dengan XAMPP

Jalankan XAMPP Control Panel hingga muncul jendela aplikasi XAMPP Control Panel. Kemudian klik tombol Start pada modul Apache dan MySQL. Modul Apache digunakan untuk mengakses PHPMyAdmin pada browser sehingga lebih mudah dalam mengakses MySQL karena menggunakan GUI (Graphical User Interface). Modul MySQL digunakan untuk melayani request atau query yang diterima dari PHPMyAdmin.

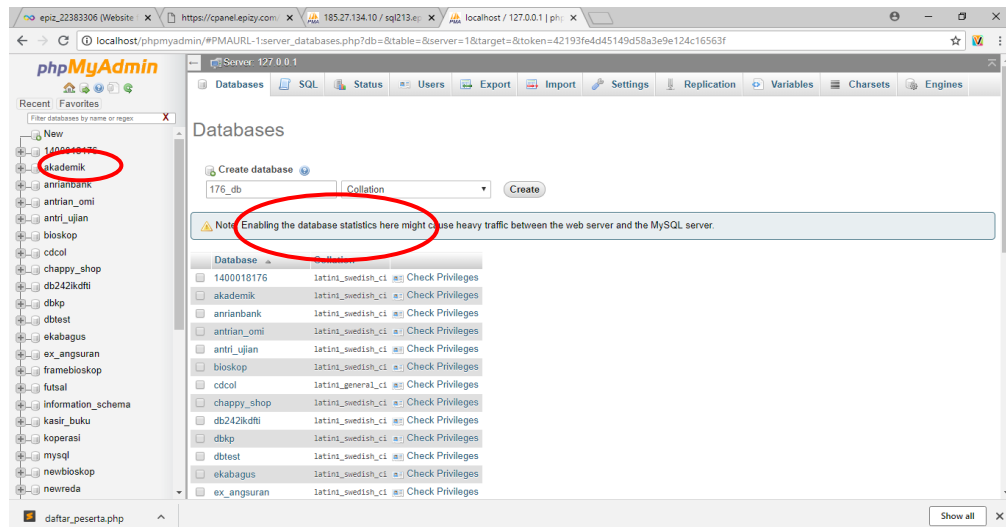


Kemudian membuka browser (Chrome, Mozilla, Opera, dll) dan mengaksesnya dengan mengetikkan “localhost/phpmyadmin” pada kolom isian URL, sehingga muncul seperti pada gambar di baawah ini.



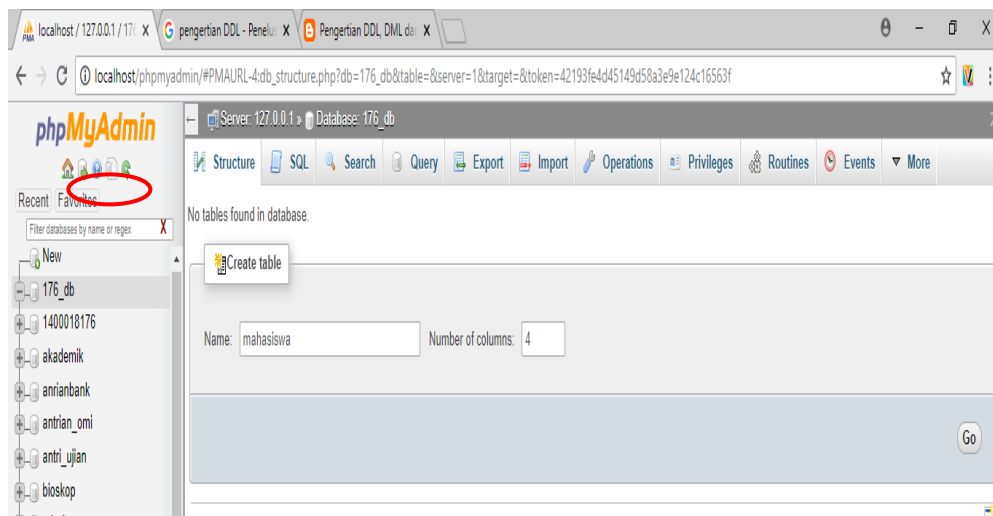
2. Membuat Database

Pilih menu “new” kemudian isi nama database pada kolom yang sudah disediakan, untuk keseragaman nama database diisi dengan “3_digit_nim_terakhir_db” kemudian pilih “create”.



3. Membuat Tabel

Membuat tabel dengan nama tabel “mahasiswa” yang memiliki atribut nim, nama, program studi dan alamat. Langkahnya yaitu pilih menu database yang hingga muncul gambar seperti di bawah ini.



Kemudian mengisi form untuk memberi atribut pada tabel “mahasiswa” yang sudah dibuat. Pada entitas ini, atribut nim akan digunakan sebagai pembeda pada data yang akan dimasukkan ke dalam tabel atau sering disebut **primary key**. Atribut nim, nama dan program studi diidentifikasi dengan tipe VARCHAR dengan panjang karakter yang disesuaikan dengan keinginan programmer. Sedangkan atribut alamat diidentifikasi dengan tipe TEXT karena memiliki jumlah karakter yang lebih panjang.

Table name: mahasiswa Add 1 column(s) Go

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	A_I	Com
nim	VARCHAR	10	None			<input type="checkbox"/>	---	<input type="checkbox"/>	<input type="checkbox"/>
nama	VARCHAR	50	None			<input type="checkbox"/>	---	<input type="checkbox"/>	<input type="checkbox"/>
prodi	VARCHAR	50	None			<input type="checkbox"/>	---	<input type="checkbox"/>	<input type="checkbox"/>
alamat	TEXT		None			<input type="checkbox"/>	---	<input type="checkbox"/>	<input type="checkbox"/>

Table comments:

Storage Engine: InnoDB Collation:

PARTITION definition:

Kemudian pada atribut nim, karena sebagai primary key maka pada index diisi dengan PRIMARY kemudian pilih “Save”.

Table name: mahasiswa Add 1 column(s) Go

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	A_I	Com
nim	VARCHAR	10	None			<input type="checkbox"/>	PRIMARY	<input type="checkbox"/>	<input type="checkbox"/>
nama	VARCHAR	50	None			<input type="checkbox"/>	---	<input type="checkbox"/>	<input type="checkbox"/>
prodi	VARCHAR	50	None			<input type="checkbox"/>	---	<input type="checkbox"/>	<input type="checkbox"/>
alamat	TEXT		None			<input type="checkbox"/>	---	<input type="checkbox"/>	<input type="checkbox"/>

Table comments:

Storage Engine: InnoDB Collation:

PARTITION definition:

Pengolahan Data

Pertemuan ke : 2

Alokasi Waktu : 2,5 jam (150 menit)

Kompetensi Dasar :

1. Praktikan mampu mengenal MySQL dan *Database Management System*.
2. Praktikan mampu memahami tabel, tipe data dan primary key dalam *Database Management System*.
3. Praktikan mampu mengidentifikasi entitas pada suatu kasus yang disediakan.
4. Praktikan mampu mengolah data pada tabel yang meliputi menambah data, menghapus data dan mengedit data.

Indikator : Insert, Update, Delete

A. Pendahuluan

Basis Data adalah suatu kumpulan data terhubung yang disimpan secara bersama-sama pada suatu media, yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu, dan dengan *software* untuk melakukan manipulasi untuk kegunaan tertentu.

Operasi dasar basis data :

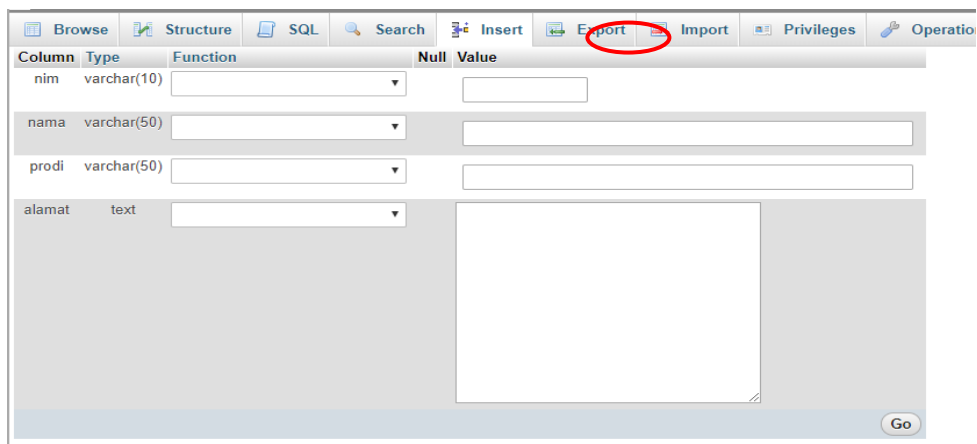
- a. Create database
- b. Drop database
- c. Create table
- d. Drop table
- e. Insert
- f. Retrieve / Search
- g. Update
- h. Delete

B. Petunjuk Praktikum

1. Insert Data

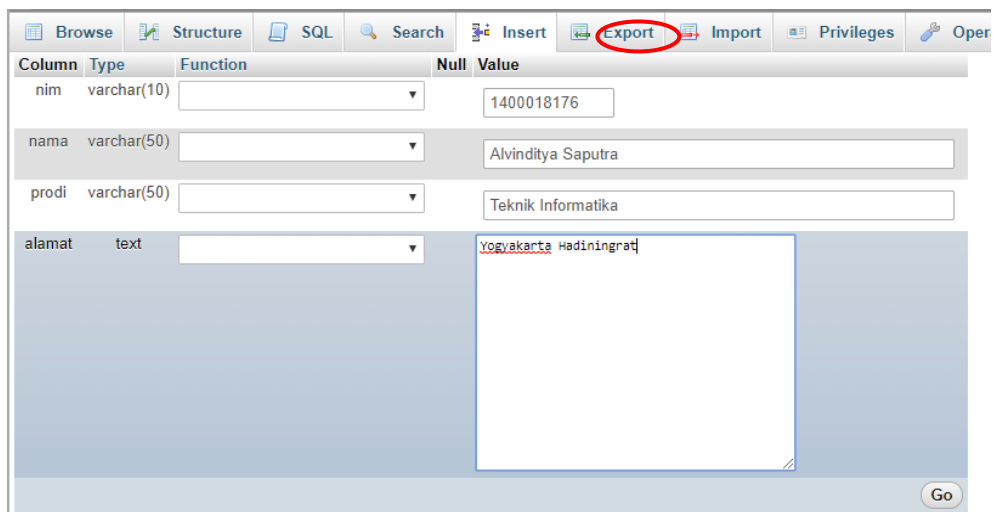
Pada langkah ini digunakan untuk menginputkan data ke dalam tabel “mahasiswa”, sebagai contoh akan diinputkan sebanyak 3 data. Langkah menambahkan data adalah sebagai berikut :

- Pilih menu “Insert” sehingga muncul seperti pada gambar di bawah ini.



The screenshot shows the 'Insert' menu in a database application. The 'Export' button is circled in red. The interface includes a toolbar with 'Browse', 'Structure', 'SQL', 'Search', 'Insert', 'Export', 'Import', 'Privileges', and 'Operations'. Below the toolbar, there is a table with columns 'Column', 'Type', 'Function', 'Null', and 'Value'. The table has four rows: 'nim' (varchar(10)), 'nama' (varchar(50)), 'prodi' (varchar(50)), and 'alamat' (text). Each row has a corresponding input field in the 'Value' column. A 'Go' button is located at the bottom right of the form.

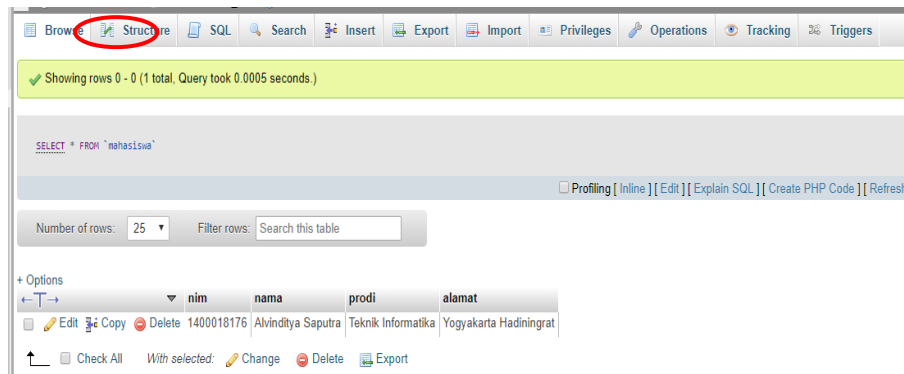
- Kemudian isi field atau kolom isian pada bagian value kemudian pilih “Go”, seperti pada gambar di bawah ini.



The screenshot shows the 'Insert' menu in a database application. The 'Export' button is circled in red. The input fields are filled with data: 'nim' is 1400018176, 'nama' is Alvinditya Saputra, 'prodi' is Teknik Informatika, and 'alamat' is Yogyakarta Hadiningrat. A 'Go' button is located at the bottom right of the form.



- c. Untuk memeriksa apakah data sudah masuk ke dalam database, dapat dilakukan dengan cara pilih pada menu "Browse". Seperti pada gambar di bawah ini.



Ulangi langkah menginputkan data hingga terdapat 3 data dalam tabel "mahasiswa". Perlu diketahui bahwa atribut nim sebagai primary key, sehingga pada value yang diinputkan tidak boleh sama.

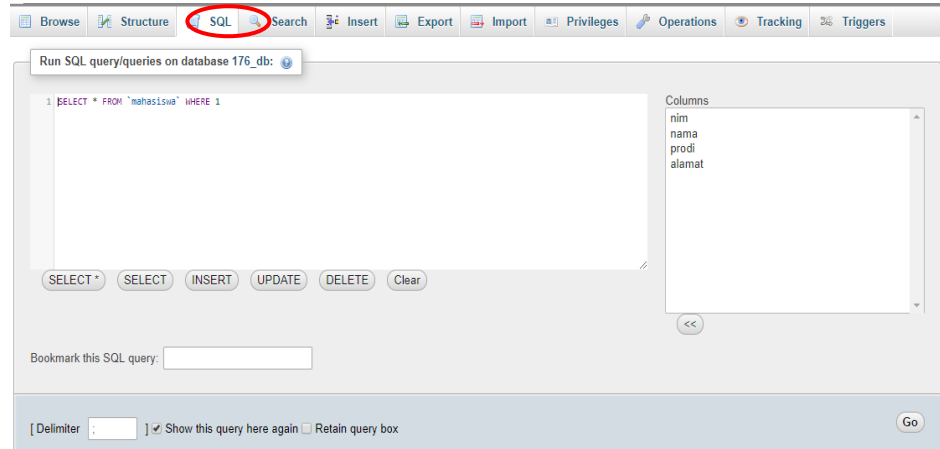
2. Read Data

Pada langkah ini digunakan untuk membaca atau mengambil data yang terdapat pada tabel mahasiswa. Terdapat beberapa query untuk membaca data sesuai yang ingin ditampilkan.

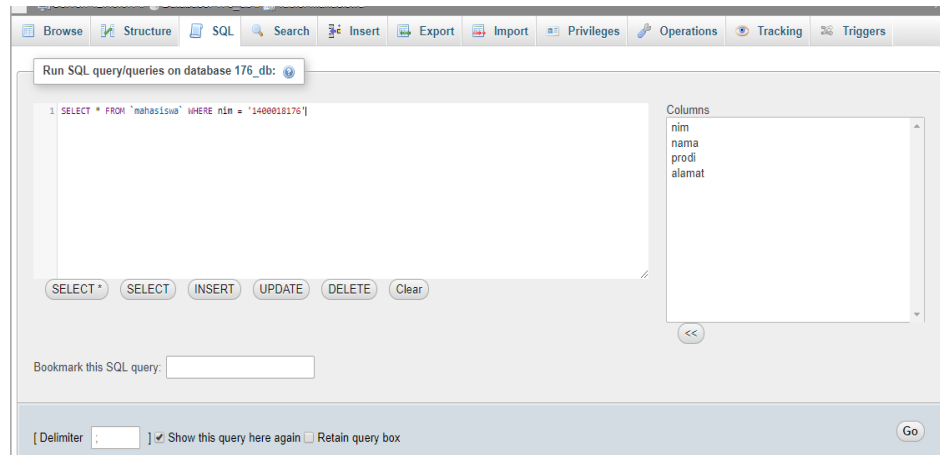
- SELECT *** : Untuk membaca data dengan seluruh atribut yang ditampilkan. Query : **SELECT * FROM nama_tabel.**
- SELECT** : Untuk membaca data dengan satu atau lebih atribut yang ditampilkan. Query : **SELECT atribut_1, atribut_2 FROM nama_tabel.**
- SELECT ... WHERE** : Untuk membaca data dengan suatu kondisi yang diinginkan. Query : **SELECT * FROM nama_tabel WHERE id = 1.**

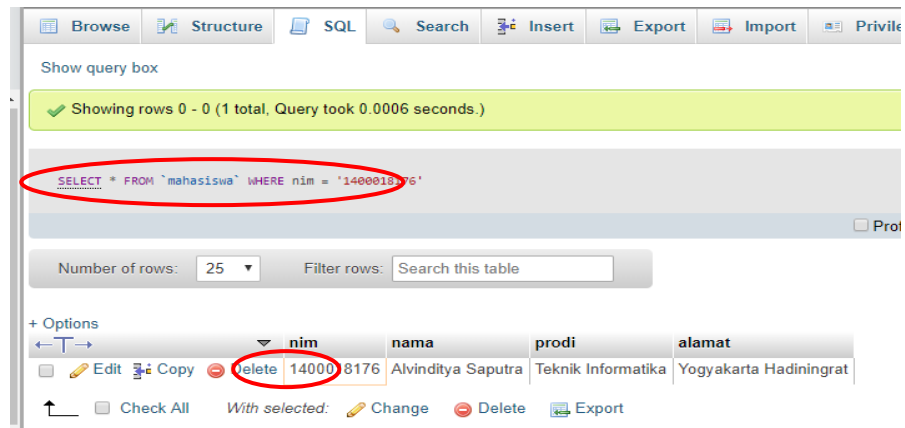
Langkah menampilkan data adalah sebagai berikut :

- a. Pilih menu SQL pada PHPMysql sehingga seperti pada gambar dibawah ini.



- b. Misalkan data yang akan ditampilkan yaitu mahasiswa dengan nim 1400018176. Maka query yang dibutuhkan yaitu **SELECT * FROM mahasiswa WHERE nim = '1400018176'**. Kemudian pilih “Go”, sehingga tampak seperti pada gambar di bawah ini.

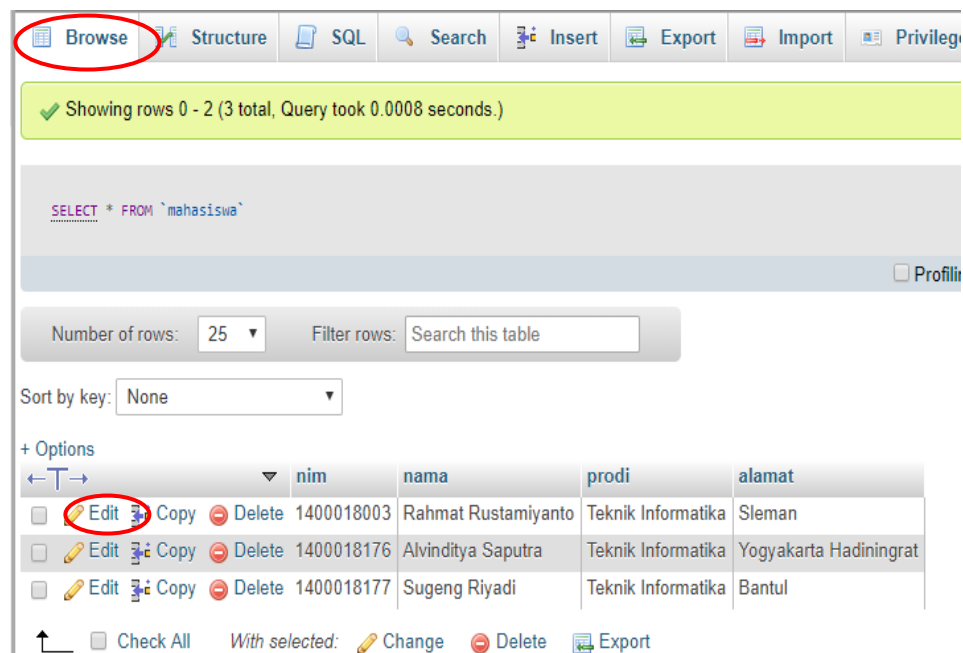




3. Update Data

Pada langkah ini digunakan untuk mengubah atau memodifikasi nilai atribut atau data dalam tabel atau entitas. Langkah mengubah data atau *update data* adalah sebagai berikut :

- Pilih menu “Browse” kemudian klik “edit” pada value atau field yang akan diubah datanya. Seperti pada gambar di bawah ini.



- b. Kemudian isi field yang akan diubah, misalkan data yang akan diubah yaitu pada field alamat. Kemudian klik “Go” sehingga seperti pada gambar dibawah ini.

Column	Type	Function	Null	Value
nim	varchar(10)			1400018003
nama	varchar(50)			Rahmat Rustamiyanto
prodi	varchar(50)			Teknik Informatika
alamat	text			Sleman Setelah diubah datanya

Go

4. Delete Data

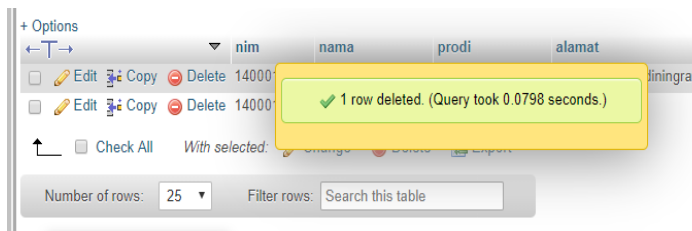
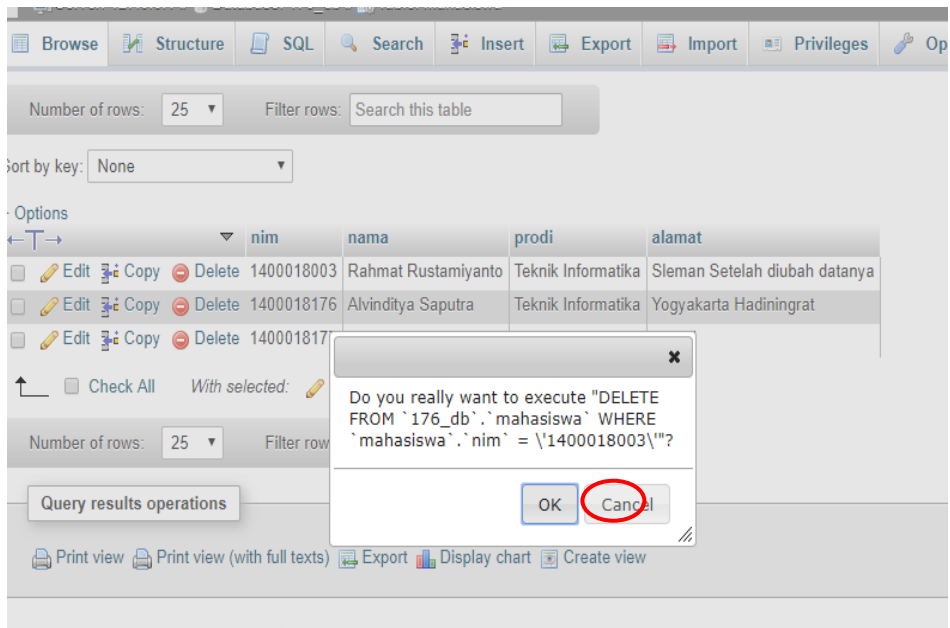
Pada langkah ini perintah DELETE digunakan untuk menghapus satu atau lebih data dalam sebuah tabel. Langkah menghapus data yaitu dengan klik “Delete” pada menu “Browse”. Seperti pada gambar dibawah ini.

Number of rows: 25 Filter rows: Search this table

Sort by key: None

+ Options

	nim	nama	prodi	alamat
<input type="checkbox"/> Edit Copy Delete	1400018003	Rahmat Rustamiyanto	Teknik Informatika	Sleman Setelah diubah datanya
<input type="checkbox"/> Edit Copy Delete	1400018176	Alvinditya Saputra	Teknik Informatika	Yogyakarta Hadiningrat
<input type="checkbox"/> Edit Copy Delete	1400018177	Sugeng Riyadi	Teknik Informatika	Bantul



Relasi Tabel

Pertemuan ke : 3

Alokasi waktu : 2,5 jam (150 menit)

Kompetensi dasar :

1. Praktikan mampu memahami konsep dasar relasi antar entitas pada basis data.
2. Praktikan mampu membuat relasi satu atau lebih tabel dalam basis data.
3. Praktikan mampu mengintegrasikan Foreign Key dengan relasional.

Indikator : Relational, Entity

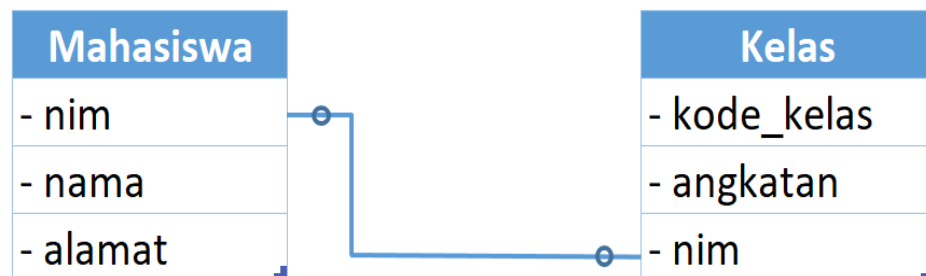
A. Pendahuluan

1. Relasi

Hubungan antara tabel yang mempresentasikan hubungan antar objek di dunia nyata. Relasi merupakan hubungan yang terjadi pada suatu tabel dengan lainnya yang mempresentasikan hubungan antar objek di dunia nyata dan berfungsi untuk mengatur mengatur operasi suatu database. Hubungan yang dapat dibentuk dapat mencakup 3 macam hubungan, yaitu :

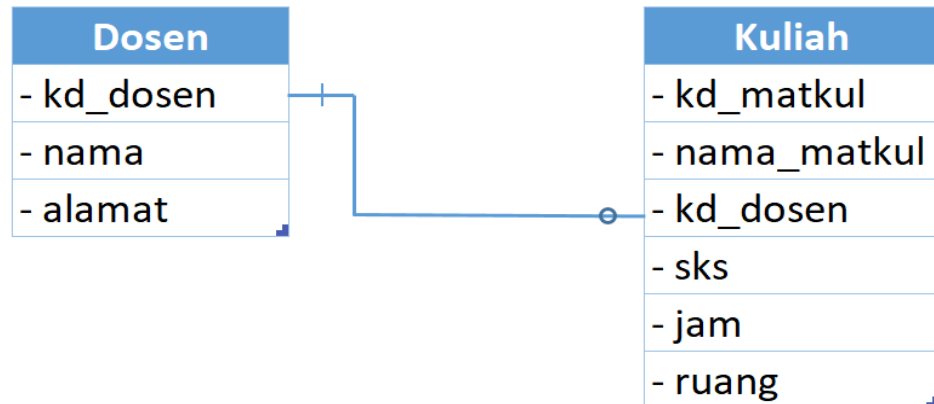
a. One to One

Mempunyai pengertian "Setiap baris data pada tabel pertama dihubungkan hanya ke satu baris data pada tabel ke dua". Contohnya : Mahasiswa menjadi ketua kelas di setiap kelasnya.



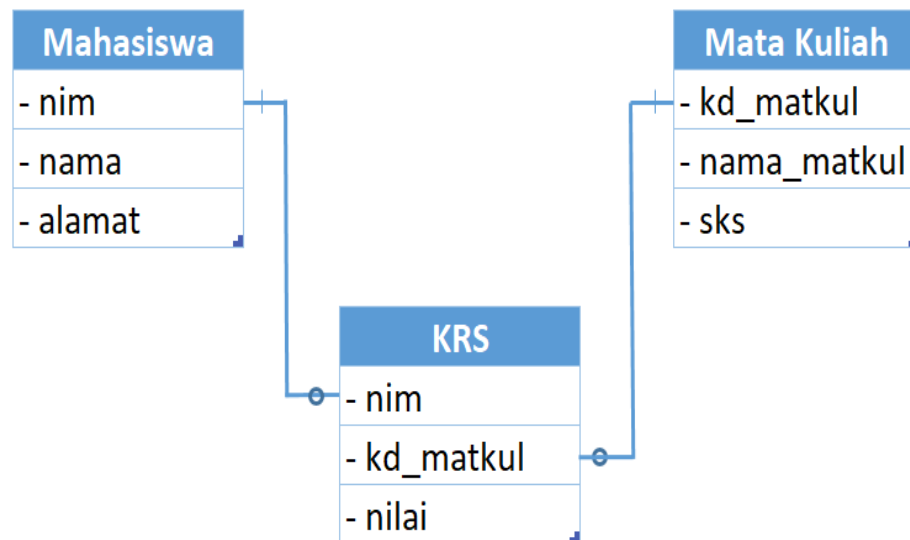
b. One to Many

Mempunyai pengertian "Setiap baris data dari tabel pertama dapat dihubungkan ke satu baris atau lebih data pada tabel ke dua". Contohnya : Dosen mengajar banyak mata kuliah.



c. Many to Many

Mempunyai pengertian "Satu baris atau lebih data pada tabel pertama bisa dihubungkan ke satu atau lebih baris data pada tabel ke dua". Artinya ada banyak baris di tabel satu dan tabel dua yang saling berhubungan satu sama lain. Contohnya : Mahasiswa mengambil mata kuliah.



2. Primary dan Foreign Key

Primary key adalah field kunci / utama dari suatu tabel yang menunjukkan bahwa field yang menjadi kunci tersebut tidak bisa diisi dengan data yang sama, / dengan kata lain Primary key menjadikan tiap record **memiliki identitas sendiri-sendiri yang membedakan satu sama lainnya (unik)**. Sedangkan **foreign key** yaitu satu atribut (atau satu set atribut) yang **melengkapi satu relationship (hubungan) yang menunjukkan ke induknya**. *Foreign key* berguna untuk mendefinisikan kolom-kolom pada suatu tabel yang nilainya mengacu ke tabel lain, jadi kolom foreign key nilainya harus diambil dari nilai kolom pada tabel lain.

B. Petunjuk Praktikum










1. Tabel Utama

Membuat tabel “Dosen” dengan atribut kode_dsn, nama_dsn, dan alamat_dsn serta membuat tabel “Mata_Kuliah” dengan atribut kode_mk, nama_mk, sks. Kemudian isi data pada masing-masing tabel sebanyak 3 data seperti pada pertemuan sebelumnya.

Tabel Dosen :

	kode_dsn	nama_dsn	alamat_dsn
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	50001	Slamet Widodo	Yogyakarta
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	50002	Wahyu Subrata	Bandung
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	50003	Sri Linggajati	Lampung

Tabel Mata_Kuliah :

<div>↩ T ↪</div>				kode_mk	nama_mk	sks	
<input type="checkbox"/>		Edit	 Copy	 Delete	T1001	Basis Data	4
<input type="checkbox"/>		Edit	 Copy	 Delete	T1002	Pemrograman Web	3
<input type="checkbox"/>		Edit	 Copy	 Delete	T1003	Statistika	4

2. Tabel Relasi One to One

Membuat tabel “Kelas” untuk mengimplementasikan jenis relasi One to One, seperti pada gambar di bawah ini.

Table name: Add column(s)

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index
<input type="text" value="kode_kelas"/>	<input type="text" value="VARCHAR"/>	<input type="text" value="10"/>	<input type="text" value="None"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	<input type="text" value="PRIMARY"/>
<input type="text" value="angkatan"/>	<input type="text" value="VARCHAR"/>	<input type="text" value="50"/>	<input type="text" value="None"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	<input type="text" value=""/>
<input type="text" value="nim_ketua"/>	<input type="text" value="VARCHAR"/>	<input type="text" value="10"/>	<input type="text" value="None"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	<input type="text" value=""/>
<input type="text"/>	<input type="text" value="INT"/>	<input type="text"/>	<input type="text" value="None"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	<input type="text" value=""/>

Kemudian menginputkan data dengan catatan nim harus sudah ada pada tabel mahasiswa, karena tabel ini akan membaca informasi data dari tabel mahasiswa. Seperti pada contoh di bawah ini.

+ Options

			kode_kelas	angkatan	nim_ketua	
<input type="checkbox"/>				14_C	2014	1400018176

With selected:
 Change
 Delete
 Export

PHP dan MySQL

Pertemuan ke : 4

Alokasi waktu : 2,5 jam (150 menit)

Kompetensi dasar :

1. Praktikan mampu membuat koneksi pada database menggunakan PHP Objek.
2. Praktikan mampu membuat query CRUD pada database menggunakan PHP Objek.

Indikator : MySQL, PHP, HTML

A. Pendahuluan

PHP adalah bahasa pemrograman script server-side yang didesain untuk pengembangan web. Selain itu, PHP juga bisa digunakan sebagai bahasa pemrograman umum. PHP disebut bahasa pemrograman **server side** karena PHP diproses pada komputer server. Hal ini berbeda dibandingkan dengan bahasa pemrograman client-side seperti JavaScript yang diproses pada web browser (client).

Objek adalah sekumpulan software yang terdiri dari variable dan method-method yang terkait. Objek juga merupakan benda nyata yang di buat berdasarkan rancangan yang di definisikan di dalam class. Object adalah instance dari class. Jika class secara umum mepresentasikan (template) sebuah object, sebuah instance adalah representasi nyata dari class itu sendiri.

Class adalah prototype, atau blueprint, atau rancangan yang mendefinisikan variable dan method-methode pada seluruh objek tertentu. Class berfungsi untuk menampung isi dari program yang akan di jalankan, di dalamnya berisi atribut / type data dan method untuk menjalankan suatu program. Sedangkan method merupakan suatu operasi berupa fungsi-fungsi yang dapat dikerjakan oleh suatu object. Method didefinisikan pada class akan tetapi dipanggil melalui object.

B. Petunjuk Praktikum

1. Membuat Class pada PHP

Membuat folder yang berada pada “C:\xampp\htdocs\” yang digunakan untuk mengakses file php didalamnya melalui web browser. Membuat file php yang akan digunakan sebagai tempat pembuatan class dalam PHP, misalkan dengan nama ‘database.php’.

<?php
Class Database {
function __construct(){
\$this->db = new mysqli("localhost", "root", "", "176_db");
}
}
?>

2. Menampilkan Data Mahasiswa

Membuat function yang digunakan untuk menampilkan data mahasiswa.

function tampilMhs () {
\$array = array();
\$query = \$this->db->query("SELECT * FROM mahasiswa");
while (\$data = mysqli_fetch_array(\$query)) {
\$array[] = \$data;
}
return \$array;
}

Kemudian membuat file index.php yang akan digunakan untuk menampilkan data dari function tampilMhs ke dalam web.

<?php
require 'database.php';
\$objek = new Database();
\$dataMhs = \$objek->tampilMhs();
?>

Source code diatas berfungsi untuk memanggil file database.php kemudian dibuat objek dari class Database. Kemudian membuat source code HTML dan PHP untuk menampilkan hasil dari data yang akan ditampilkan.

<table border="1">
<tr>
<th>NIM</th>
<th>Nama</th>
<th>Program Studi</th>
<th>Alamat</th>
</tr>
<?php foreach (\$dataMhs as \$x) {?>
<tr>
<td><?php echo \$x['nim'];?></td>
<td><?php echo \$x['nama'];?></td>
<td><?php echo \$x['prodi'];?></td>
<td><?php echo \$x['alamat'];?></td>
</tr>
<?php } ?>
</table>

Fungsi foreach digunakan untuk memecah data array yang dikembalikan oleh function tampilMhs pada class Database.

NIM	Nama	Program Studi	Alamat
1400018176	Alvinditya Saputra	Teknik Informatika	Yogyakarta Hadiningrat
1400018177	Sugeng Riyadi	Teknik Informatika	Bantul

3. Menambah Data Mahasiswa

Hal yang harus dilakukan terlebih dahulu yaitu membuat form input mahasiswa dengan menambahkan source code dibawah ini pada index.php.

<form method="POST" action="">
<table>
<tr>
<td>NIM</td>
<td>:</td>
<td><input type="text" name="nim"></td>
</tr>
<tr>
<td>Alamat</td>
<td>:</td>
<td><input type="text" name="nama"></td>
</tr>
<tr>
<td>Program Studi</td>
<td>:</td>
<td><input type="text" name="prodi"></td>
</tr>
<tr>
<td>Alamat</td>
<td>:</td>

<td><input type="text" name="alamat"></td>
</tr>
<tr><td colspan="3" align="right"><input type="submit" name="simpan" value="SIMPAN"></td></tr>
</table>
</form>

NIM	Nama	Program Studi	Alamat
1400018176	Alvinditya Saputra	Teknik Informatika	Yogyakarta Hadiningrat
1400018177	Sugeng Riyadi	Teknik Informatika	Bantul

NIM :

Alamat :

Program Studi :

Alamat :

Kemudian membuat function pada class database yang berfungsi untuk menginputkan data ke dalam database.

function tambahMhs (\$nim, \$nama, \$prodi, \$alamat){
\$insert = \$this->db->query("INSERT INTO mahasiswa (nim, nama, prodi, alamat) VALUES ('\$nim','\$nama','\$prodi','\$alamat')");
if (\$insert) {
header('location:index.php');
} else {
echo "data gagal ditambahkan";

<code>echo "
\$nim \$nama \$prodi \$alamat";</code>
<code>echo \$this->db->error;</code>
<code>}</code>
<code>}</code>

Kemudian menambahkan source code pada index php yang berfungsi untuk mengantarkan data pada function yang sudah dibuat pada langkah sebelumnya.

<code><?php</code>
<code>if (isset(\$_POST['simpan'])) {</code>
<code> \$nim = \$_POST['nim'];</code>
<code> \$nama = \$_POST['nama'];</code>
<code> \$prodi = \$_POST['prodi'];</code>
<code> \$alamat = \$_POST['alamat'];</code>
<code> \$objek->tambahMhs(\$nim, \$nama, \$prodi, \$alamat);</code>
<code>}</code>
<code>?></code>

NIM	Nama	Program Studi	Alamat
1400018176	Alvinditya Saputra	Teknik Informatika	Yogyakarta Hadiningrat
1400018177	Sugeng Riyadi	Teknik Informatika	Bantul

NIM :

Alamat :

Program Studi :

Alamat :

NIM	Nama	Program Studi	Alamat
1400018176	Alvinditya Saputra	Teknik Informatika	Yogyakarta Hadiningrat
1400018177	Sugeng Riyadi	Teknik Informatika	Bantul
1400018182	Andi Eko	Teknik Informatika	Lampung

NIM :

Alamat :

Program Studi :

Alamat :

4. Mengubah dan Menghapus Data Mahasiswa

Mengubah syntak html pada index.php yang menampilkan tabel data mahasiswa, seperti di bawah ini.

<tr>
<th>NIM</th>
<th>Nama</th>
<th>Program Studi</th>

<th>Alamat</th>
<th>Aksi</th>
</tr>
<?php foreach (\$dataMhs as \$x) {?>
<tr>
<td><?php echo \$x['nim'];?></td>
<td><?php echo \$x['nama'];?></td>
<td><?php echo \$x['prodi'];?></td>
<td><?php echo \$x['alamat'];?></td>
<td><a href="edit.php?nim=<?php echo \$x['nim'];?>">Edit <a href="index.php?aksi=delete&nim=<?php echo \$x['nim'];?>">Hapus</td>
</tr>

Kemudian membuat function update pada class Database.

function updateMhs(\$nim, \$nama, \$prodi, \$alamat){
\$insert = \$this->db->query("UPDATE mahasiswa SET nama = '\$nama', prodi = '\$prodi', alamat = '\$alamat' WHERE nim = '\$nim'");
if (\$insert) {
header('location:index.php');
} else {
echo "data gagal ditambahkan";
echo " \$nim \$nama \$prodi \$alamat";
echo \$this->db->error;
}

}

Kemudian membuat file edit.php dengan syntak html dan php seperti di bawah ini.

<?php
require 'database.php';
\$objek = new Database();
\$mhs = \$objek->tampilMhsWhere(\$_GET['nim']);
foreach (\$mhs as \$x) {
}
if (isset(\$_POST['simpan'])) {
\$nim = \$_POST['nim'];
\$nama = \$_POST['nama'];
\$prodi = \$_POST['prodi'];
\$alamat = \$_POST['alamat'];
\$objek->updateMhs(\$nim, \$nama, \$prodi, \$alamat);
}
?>
<form method="POST" action="">
<table>
<tr>
<td>NIM</td>
<td>:</td>
<td><input type="text" name="nim"
value="<?php echo \$x['nim'];?>" readonly></td>
</tr>
<tr>
<td>Alamat</td>

<td>:</td>
<td><input type="text" name="nama" value="<?php echo \$x['nama'];?>"></td>
</tr>
<tr>
<td>Program Studi</td>
<td>:</td>
<td><input type="text" name="prodi" value="<?php echo \$x['prodi'];?>"></td>
</tr>
<tr>
<td>Alamat</td>
<td>:</td>
<td><input type="text" name="alamat" value="<?php echo \$x['alamat'];?>"></td>
</tr>
<tr><td colspan="3" align="right"><input type="submit" name="simpan" value="SIMPAN"></td></tr>
</table>
</form>

NIM	Nama	Program Studi	Alamat	Aksi
1400018176	Alvinditya Saputra	Teknik Informatika	Yogyakarta Hadiningrat	Edit Hapus
1400018177	Sugeng Riyadi	Teknik Informatika	Bantul	Edit Hapus
1400018182	Andi Eko	Teknik Informatika	Lampung	Edit Hapus

NIM :

Alamat :

Program Studi :

Alamat :

NIM	Nama	Program Studi	Alamat	Aksi
1400018176	Alvinditya Saputra	Teknik Informatika	Yogyakarta Hadiningrat	Edit Hapus
1400018177	Sugeng Riyadi	Teknik Informatika	Bantul	Edit Hapus
1400018182	Andi Eko Suryanto	Teknik Informatika	Lampung	Edit Hapus

NIM :

Alamat :

Program Studi :

Alamat :

Kemudian membuat function hapus pada class database yang berfungsi untuk menghapus data berdasarkan nim.

```

function hapusMhs($nim){
    $insert = $this->db->query("DELETE FROM
mahasiswa WHERE nim = '$nim'");
    if ($insert) {
        header('location:index.php');
    }else{
        echo "data gagal dihapus";
    }
}

```


<code>echo "
\$nim";</code>
<code>echo \$this->db->error;</code>
<code>}</code>
<code>}</code>

Kemudian menambahkan source code php pada index.php seperti di bawah ini.

<code>if (isset(\$_GET['aksi'])) {</code>
<code>if (\$_GET['aksi'] == 'delete') {</code>
<code>\$objek->deleteMhs(\$_GET['nim']);</code>
<code>}</code>
<code>}</code>

NIM	Nama	Program Studi	Alamat	Aksi
1400018176	Alvinditya Saputra	Teknik Informatika	Yogyakarta Hadiningrat	Edit Hapus
1400018177	Sugeng Riyadi	Teknik Informatika	Bantul	Edit Hapus
1400018182	Andi Eko Suryanto	Teknik Informatika	Lampung	Edit Hapus

NIM :

Alamat :

Program Studi :

Alamat :

NIM	Nama	Program Studi	Alamat	Aksi
1400018176	Alvinditya Saputra	Teknik Informatika	Yogyakarta Hadiningrat	Edit Hapus
1400018177	Sugeng Riyadi	Teknik Informatika	Bantul	Edit Hapus

DDL dan DML

- Pertemuan ke** : 5
Alokasi Waktu : 2,5 Jam
Kompetensi Dasar : Praktikan memahami perintah DDL dan DML
Indikator : Create, Alter, Drop, Insert, Update, Delete, Select

A. Pendahuluan

Data Definition Language memiliki fungsi untuk melakukan hal-hal berikut :

1. Membuat/menghapus database, dinyatakan dengan perintah CREATE DATABASE dan DROP DATABASE
2. Membuat/menghapus table, dinyatakan dengan perintah CREATE TABLE dan DROP TABLE
3. Memodifikasi table, dinyatakan dengan perintah ALTER TABLE

Sedangkan Data Manipulation Language terdiri atas :

1. Mengisi tabel dengan data, dinyatakan dengan perintah INSERT
2. Mengedit data pada tabel, dinyatakan dengan perintah UPDATE
3. Menghapus data pada tabel, dinyatakan dengan perintah DELETE
4. Mencari data pada tabel, dinyatakan dengan perintah SELECT

B. Petunjuk Praktikum

1. Perintah DDL

1. Menampilkan database yang ada pada mysql
SHOW databases;
2. Membuat database
CREATE database <nama_database>;
CREATE database akademik;

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql      |
| test       |
+-----+
3 rows in set (0.02 sec)

mysql> create database akademik;
Query OK, 1 row affected (0.00 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| akademik         |
| mysql            |
| test             |
+-----+
4 rows in set (0.00 sec)
```

3. Menggunakan database

USE <nama_database>;

USE akademik;

4. Menghapus database

DROP <nama_database>;

DROP akademik;

```
+-----+
| Database |
+-----+
| information_schema |
| akademik         |
| mysql            |
| test             |
+-----+
4 rows in set (0.00 sec)

mysql> use akademik;
Database changed
mysql> drop database akademik;
Query OK, 0 rows affected (0.02 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql            |
| test             |
+-----+
3 rows in set (0.00 sec)
```

5. Melihat tabel apa yang sudah ada dalam database yang aktif

SHOW tables;

```
mysql> use akademik;
Database changed
mysql> show tables;
Empty set (0.00 sec)
```

6. Membuat tabel

CREATE TABLE <nama_tabel> (<nama_kolom><tipedata>);

CREATE TABLE mhs (NIM CHAR (8) NOT NULL PRIMARY KEY,
nama_mhs VARCHAR (30), alamat VARCHAR (35));

PRIMARY KEY adalah kunci utama, dalam setiap tabel harus ada minimal satu kolom yang dijadikan sebagai PRIMARY KEY.

NOT NULL artinya setiap kolom tidak boleh kosong, sedangkan jika diijinkan untuk dikosongkan dapat menggunakan parameter NULL.

7. Melihat struktur tabel

DESCRIBE <nama_tabel>;

DESC <nama_tabel>;

DESCRIBE mhs;

DESC mhs;

```
mysql> create table mhs (nim char(8)not null primary key, nama__mhs varchar(30),
alamat varchar(35));
Query OK, 0 rows affected (0.08 sec)

mysql> desc;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MySQL server version for the right syntax to use near '' at
line 1
mysql> desc mh;
ERROR 1146 (42S02): Table 'akademik.mh' doesn't exist
mysql> desc mhs;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nim   | char(8) | NO | PRI | NULL | |
| nama__mhs | varchar(30) | YES | | NULL | |
| alamat | varchar(35) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> _
```

8. Mengubah struktur tabel

Ada saatnya kita sadar kalo ternyata struktur tabel yang pernah dibuat perlu penyempurnaan, bisa dalam hal penambahan kolom, pengubahan lebar kolom, penghapusan kolom, dan sebagainya, sehingga dengan penggunaan perintah

ALTER ini maka kita dapat mengubah kekurangan atau kesalahan pada saat kita membuat tabel.

ALTER TABEL nama_tabel [spesifikasi perubahan]

Parameter [spesifikasi perubahan] adalah pilihan yang digunakan untuk mengubah struktur tabel yaitu CHANGE, ADD, DROP.

ALTER TABLE nama_tabel jenis_pengubahan

a. Menambah kolom baru

Parameter yang digunakan adalah ADD.

ALTER TABLE nama_tabel ADD kolom_baru tipe(panjang) [FIRST | AFTER kolom_lama]

FIRST artinya kolom yang baru akan ditambahkan pada urutan pertama.

AFTER artinya kita meletakkan kolom yang baru setelah kolom yang ditunjuk.

Misalnya kita ingin menambah kolom baru yaitu no_telp pada tabel mhs dengan posisi setelah kolom/field alamat dengan tipe INT :

ALTER TABLE mhs ADD no_telp INT AFTER alamat;

```
mysql> alter table mhs add no_telp int after alamat;
Query OK, 0 rows affected (0.19 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc mhs;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nim   | char(8) | NO | PRI | NULL | |
| nama_mhs | varchar(30) | YES | | NULL | |
| alamat | varchar(35) | YES | | NULL | |
| no_telp | int(11) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.02 sec)
```

b. Mengubah nama kolom

Parameter yang digunakan adalah CHANGE.

ALTER TABEL nama_tabel CHANGE kolom_lama kolom_baru tipe (panjang)

ALTER TABLE mhs CHANGE alamat alamat_mhs VARCHAR (35)

```
mysql> alter table mhs change alamat alamat_mhs varchar(35);
Query OK, 0 rows affected (0.25 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc mhs;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nim        | char(8)       | NO   | PRI | NULL    |       |
| nama_mhs   | varchar(30)   | YES  |     | NULL    |       |
| alamat_mhs | varchar(35)   | YES  |     | NULL    |       |
| no_telp    | int(11)       | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

c. Mengubah nama table

Parameter yang digunakan adalah RENAME.

ALTER TABLE nama_lama RENAME [TO] nama_baru;

ALTER TABLE mhs RENAME TO mahasiswa;

```
mysql> alter table mhs rename to mahasiswa;
Query OK, 0 rows affected (0.06 sec)

mysql> show tables;
+-----+
| Tables_in_akademik |
+-----+
| mahasiswa           |
+-----+
1 row in set (0.00 sec)
```

d. Menghapus atau menghilangkan komponen pada tabel

Menghapus ini dapat mencakup menghilangkan primary key, kolom, tabel, index pada kolom.

Untuk menghilangkan PRIMARY KEY

ALTER TABLE nama_tabel DROP PRIMARY KEY;

Untuk menghilangkan salah satu kolom pada tabel

ALTER TABLE mahasiswa DROP no_telp;

```
mysql> alter table mahasiswa drop no_telp;
Query OK, 0 rows affected (0.17 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc mahasiswa;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nim        | char(8)       | NO   | PRI | NULL    |       |
| nama_mhs   | varchar(30)   | YES  |     | NULL    |       |
| alamat_mhs | varchar(35)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

II. Perintah DML

1. Mengisi tabel

Ada beberapa cara dalam memasukkan data yaitu dengan menyamakan kolom dan data, menyebutkan kolom, tanpa menyebutkan kolom, memasukkan hanya sebagai pada kolom.

Menyamakan Kolom dan Data, perintahnya sebagai berikut:

```
INSERT INTO nama_tabel SET
```

```
kolom_pertama = 'data_kolom_pertama',
```

```
kolom_kedua = 'data_kolom_kedua',
```

```
kolom_terakhir = 'data kolom terakhir' ;
```

```
INSERT INTO mhs SET NIM = '11010010', nama_mhs = 'M Rauf',
alamat_mhs='Puri Kencana B2 Sleman';
```

Menyebutkan Kolom, perintahnya sebagai berikut:

```
INSERT INTO nama_tabel (kolom_pertama, kolom_kedua, kolom_terakhir)
```

```
VALUES (data_kolom_pertama, data_kolom_kedua, data_kolom_terakhir);
```

```
INSERT INTO mhs (NIM, nama_mhs, alamat_mhs)
```

```
VALUES ('11010010', 'M Rauf', 'Puri Kencana B2 Sleman')
```

Tanpa Menyebutkan Kolom

INSERT INTO nama_tabel VALUES (data_kolom_pertama,
data_kolom_kedua,data_kolom_terakhir);

INSERT INTO mhs VALUES ('11010010', 'M Rauf', 'Puri Kencana B2 Sleman')

```
mysql> insert into mhs values ('11010010', 'M Rauf', 'Puri Kencana B2 Sleman');  
Query OK, 1 row affected (0.03 sec)  
  
mysql> insert into mhs values ('11010011', 'Rafifah Azzahra', 'Blunyah Gede 227 J  
ogja');  
Query OK, 1 row affected (0.33 sec)
```

2. Melihat isi tabel

Perintah ini digunakan untuk menyeleksi atau memilih atau menampilkan data-data yang ada dalam tabel. baik menampilkan semua kolom, sebagian kolom, serta berdasarkan kondisi.

- Perintah untuk menampilkan data apa adanya, tanpa syarat, tanpa pemilihan kolom dan tanpa urutan :

SELECT * FROM <nama_tabel>

SELECT * FROM mhs;

- Membatasi jumlah record yang dibaca

Untuk membatasi record yang muncul atau untuk mencari record dengan kriteria tertentu, digunakan klausa where. Misal untuk melihat nama mahasiswa dengan nim = '11010010'

SELECT * from mhs WHERE nim = '11010010'

Melihat data mahasiswa yang bernama 'M Rauf'

SELECT * from mhs where nama_mhs = 'M Rauf'

Melihat data mahasiswa yang beralamat di Sleman

SELECT * from mhs where alamat_mhs like '%Sleman'

Pada kriteria alamat, terlihat ada penggunaan karakter '%'. Karakter ini mengandung makna, apa pun teks yang ada akan memenuhi kriteria. Jadi '%Sleman' artinya semua string yang diakhiri kata Sleman.s


```
mysql> select * from mhs;
+-----+-----+-----+
| nim      | nama_mhs | alamat_mhs |
+-----+-----+-----+
| 11010010 | M Rauf   | Puri Kencana B2 Sleman |
| 11010011 | Rafifah Azzahra | Blunyah Gede 227 Jogja |
+-----+-----+-----+
2 rows in set (0.02 sec)

mysql> select * from mhs where nim='11010010';
+-----+-----+-----+
| nim      | nama_mhs | alamat_mhs |
+-----+-----+-----+
| 11010010 | M Rauf   | Puri Kencana B2 Sleman |
+-----+-----+-----+
1 row in set (0.00 sec)
```

- c. Membatasi jumlah field yang dibaca

Untuk melihat field-field tertentu dari tabel, gantikan karakter “*” dengan nama field yang dikehendaki.

Melihat hanya nim dan nama mahasiswa

SELECT nim, nama_mhs from mhs;

3. Menampilkan data berurutan

Klausula order by digunakan untuk mengurutkan data yang diminta dengan query.

Misal diminta untuk menampilkan nim dan nama mahasiswa yangurut oleh nim secara menaik:

SELECT nim, nama_mhs from mhs order by nim asc;

Jika urut menurun:

SELECT nim, nama_mhs from mhs order by nim desc;

```
mysql> select nim, nama_mhs from mhs order by nim asc;
+-----+-----+
| nim      | nama_mhs |
+-----+-----+
| 11010010 | M Rauf   |
| 11010011 | Rafifah Azzahra |
| 11010012 | M Tedy Farhan |
| 11010013 | Shafa Dian |
| 11010014 | Austin A Cetta |
+-----+-----+
5 rows in set (0.00 sec)

mysql> select nim, nama_mhs from mhs order by nim desc;
+-----+-----+
| nim      | nama_mhs |
+-----+-----+
| 11010014 | Austin A Cetta |
| 11010013 | Shafa Dian |
| 11010012 | M Tedy Farhan |
| 11010011 | Rafifah Azzahra |
| 11010010 | M Rauf   |
+-----+-----+
5 rows in set (0.00 sec)
```

4. Mengubah isi data tabel

UPDATE nama_tabel SET

kolom_pertama = 'data_kolom_pertama',

kolom_kedua = 'data_kolom_kedua',

kolom_terakhir = 'data_kolom_terakhir',

WHERE kondisi

Penggunaan perintah UPDATE tanpa klausa WHERE mengakibatkan semua data dalam satu kolom akan diubah.

Misal akan diubah nama 'M Rauf' menjadi 'Muhammad Rauf'

UPDATE mhs SET nama_mhs = 'M Rauf' where nim = '11010010'

```
mysql> update mhs set nama_mhs = 'Muhammad Rauf' where nim='11010010';
Query OK, 1 row affected (0.34 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select nim, nama_mhs from mhs order by nim desc;
+-----+-----+
| nim   | nama_mhs |
+-----+-----+
| 11010014 | Austin A Cetta |
| 11010013 | Shafa Dian |
| 11010012 | M Tedy Farhan |
| 11010011 | Rafifah Azzahra |
| 11010010 | Muhammad Rauf |
+-----+-----+
5 rows in set (0.00 sec)
```

```
UPDATE employees
SET job_id = (SELECT job_id
              FROM employees
              WHERE employee_id = 205),
    salary = (SELECT salary
              FROM employees
              WHERE employee_id = 205)
WHERE employee_id = 114;
1 rows updated
```

5. Menghapus isi tabel

DELETE FROM nama_tabel WHERE kondisi

```
DELETE FROM departments  
WHERE department_name = 'Finance';  
1 rows deleted
```

Apabila tidak menggunakan klausa WHERE maka akan menyebabkan semua record dalam tabel terhapus. Misal akan dihapus data mahasiswa bernama 'M Rauf'

DELETE FROM mhs WHERE nim = '11010012'

Fungsi Agregasi

Pertemuan ke : 6
Alokasi Waktu : 2,5 Jam
Kompetensi Dasar : Praktikan dapat menerapkan fungsi agregasi dengan perintah SQL
Indikator : AVG, Count, Max, Min, Sum

A. Pendahuluan

Fungsi agregasi meliputi :

1. Menghitung banyak record
2. Menghitung total nilai suatu atribut
3. Menghitung rata-rata nilai atribut
4. Mencari nilai terbesar dari nilai atribut
5. Mencari nilai terkecil dari nilai atribut

Berikut bahasa query formal untuk fungsi di atas :

KLAUSA	PENJELASAN
AVG	Sama dengan
COUNT	Mengetahui jumlah record
MAX	Mengetahui nilai maximal
MIN	Mengetahui nilai minimal
SUM	Menghitung jumlah data

Berikut beberapa operator yang biasanya diikuti Klausula WHERE :

OPERATOR	PENJELASAN
=	Sama dengan
< >, !=	Tidak sama dengan

<	Kurang dari
>	Lebih besar dari
<=	Kurang dari atau sama dengan
>=	Lebih dari atau sama dengan
!>	Tidak lebih besar dari
!<	Tidak lebih kecil dari
BETWEEN	Antara dua nilai yang ditentukan
LIKE	Menyesuaikan nilai yang ditentukan
IS NULL	Nilainya adalah NULL
IN	Nilainya ditentukan dalam sebuah daftar
NOT	Negasi dari sebuah operator perbandingan
AND	Merangkai kriteria pencarian
OR	Memastikan bahwa criteria pencarian adalah eksklusif

B. Petunjuk Praktikum

Buat tabel mata_kuliah dan isikan seperti gambar di bawah :

```
mysql> select * from mata_kuliah;
+----+-----+-----+-----+
| kode_kul | nama_kul | sks | sem |
+----+-----+-----+-----+
| IT0101 | Logika Inf | 3 | 1 |
| IT0102 | Studi Islam I | 2 | 1 |
| IT0103 | Kalkulus Inf | 3 | 1 |
| IT0301 | Sertifikasi | 0 | 3 |
| IT0401 | Basis Data | 3 | 4 |
| IT0801 | Tugas Akhir | 6 | 8 |
+----+-----+-----+-----+
6 rows in set (0.00 sec)
```

1. Menampilkan data mata kuliah yang dilaksanakan di semester 1.
SELECT * FROM mata_kuliah WHERE sem = 1;
2. Menampilkan data mata kuliah yang dilaksanakan selain semester 1.
SELECT * FROM mata_kuliah WHERE sem <>1;
3. Menampilkan data mata kuliah yang mengandung judul informatika

SELECT * FROM mat_kul WHERE nama_kul LIKE '%informatika%'

```
mysql> select * from mata_kuliah where sem=1;
+----+-----+-----+-----+
| kode_kul | nama_kul | sks | sem |
+----+-----+-----+-----+
| IT0101 | Logika Inf | 3 | 1 |
| IT0102 | Studi Islam I | 2 | 1 |
| IT0103 | Kalkulus Inf | 3 | 1 |
+----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> select * from mata_kuliah where sem<>1;
+----+-----+-----+-----+
| kode_kul | nama_kul | sks | sem |
+----+-----+-----+-----+
| IT0301 | Sertifikasi | 0 | 3 |
| IT0401 | Basis Data | 3 | 4 |
| IT0801 | Tugas Akhir | 6 | 8 |
+----+-----+-----+-----+
3 rows in set (0.01 sec)

mysql> select * from mata_kuliah where nama_kul like '%Inf';
+----+-----+-----+-----+
| kode_kul | nama_kul | sks | sem |
+----+-----+-----+-----+
| IT0101 | Logika Inf | 3 | 1 |
| IT0103 | Kalkulus Inf | 3 | 1 |
+----+-----+-----+-----+
2 rows in set (0.00 sec)
```

4. Menampilkan data mata kuliah yang mengandung judul informatika dan sksnya = 3
SELECT * FROM mat_kul WHERE nama_kul LIKE '%informatika%' AND sks = 3
5. Menghitung jumlah data mata kuliah
SELECT COUNT(*) from mat_kul;
6. Menghitung sks yang paling sedikit, sks yang paling banyak dan rata-rata sks
SELECT MIN (sks), MAX (sks), AVG (sks) from mat_kul;
7. Menghitung total jumlah sks
SELECT SUM (sks) from mat_kul;

```
mysql> select count(*) from mata_kuliah;
+-----+
| count(*) |
+-----+
|         6 |
+-----+
1 row in set (0.00 sec)

mysql> select min(sks), max(sks), avg(sks) from mata_kuliah;
+-----+-----+-----+
| min(sks) | max(sks) | avg(sks) |
+-----+-----+-----+
| 0         | 6         | 2.833333333333333 |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql> select sum(sks) from mata_kuliah;
+-----+
| sum(sks) |
+-----+
|        17 |
+-----+
1 row in set (0.00 sec)
```

Query dari Relasi Tabel

Pertemuan ke : 7

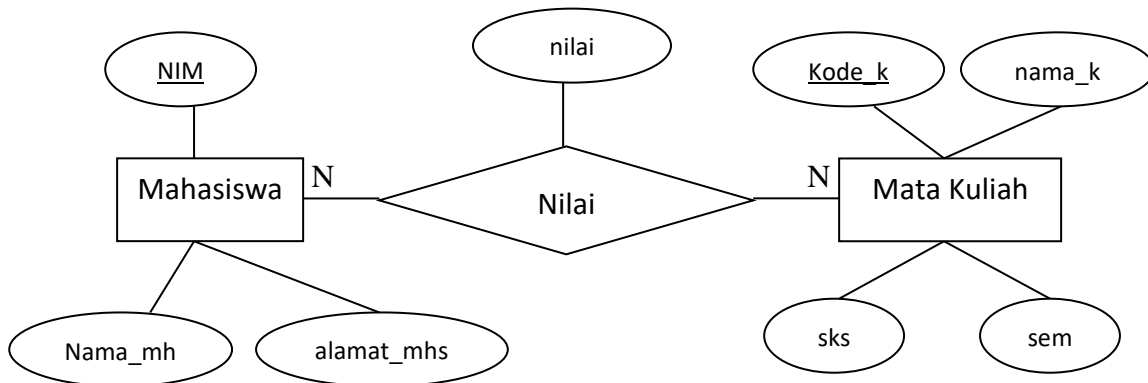
Alokasi Waktu : 2,5 Jam

Kompetensi Dasar : Praktikan dapat melakukan pencarian data antar tabel

Indikator : Relasi antar tabel

A. Pendahuluan

Pada perancangan basis data dengan ERD akan mencerminkan mapping table yang akan dibentuk pada basis data fisik. Seperti halnya ERD berikut :



Mapping table dari ERD di atas yaitu :

Mhs (nim, nama_mhs, alamat_mhs)

Mata_Kuliah (kode_kul, nama_kul, sks, sem)

Nilai (nim, kode_kul, nilai)

Query terhadap 2 tabel atau lebih tidak bisa dilakukan sembarangan. Tabel-tabel yang menjadi sumber *query* harus memiliki keterhubungan (relasi). NIM adalah primary key dari tabel mhs dan kode_kul adalah primary key dari mata_kuliah. Tabel nilai

adalah tabel yang menjadi relasi antara tabel mhs dan mata_kuliah, sehingga primary key dari tabel nilai diambil dari primary key tabel mhs dan mata_kuliah, yaitu nim dan kode_kul. Artinya nim adalah foreign key ke tabel mhs dan juga kode_kul adalah foreign key ke tabel mata_kuliah.

Perintah untuk menampilkan data dari beberapa tabel, yaitu :

SELECT

<tabel1>.<kolom_1>, <tabel1>.<kolom_2>, <tabel1>.<kolom_n>,
<tabel2>.<kolom_1>, <tabel2>.<kolom_2>, <tabel2>.<kolom_n>,
<tabeln>.<kolom_1>, <tabeln>.<kolom_2>, <tabeln>.<kolom_n>

FROM

<tabel1>, <tabel2>, <tabeln>

WHERE

<tabel1>.<kolom_x> = <tabel2>.<kolom_y> and
<tabel2>.<kolom_y> = <tabeln>.<kolom_n> and
<tabel1>.<kolom_x> = <tabeln>.<kolom_n>

Kondisi yang dituliskan pada WHERE adalah atribut yang menjadi relasi yang menghubungkan antara tabel 1, tabel 2 sampai tabel n.

Untuk contoh kasus tabel di atas, perintah WHERE dapat ditulis :

WHERE mhs.nim = nilai.nim and nilai.kode_kul = mata_kuliah.kode_kul

B. Petunjuk Praktikum

Buatlah tabel berikut :

Mhs (nim, nama_mhs, alamat_mhs)

Mata_Kuliah (kode_kul, nama_kul, sks, semester)

Nilai (nim, kode_kul, nilai)

```
mysql> show tables;
+-----+
| Tables_in_akademik |
+-----+
| mata_kuliah         |
| mhs                 |
| nilai               |
+-----+
3 rows in set (0.00 sec)
```

1. Membuat tabel nilai dengan menyatakan bahwa nim dan kode_kul adalah merupakan foreign key dari tabel mhs dan mata_kuliah.

```
mysql> create table nilai (nim varchar(8), kode_kul varchar(6), nilai int, foreign key fk_nim(nim) references mhs(nim), foreign key fk_kode_kul(kode_kul) references mata_kuliah(kode_kul));
Query OK, 0 rows affected (0.09 sec)

mysql> desc nilai;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nim   | varchar(8) | YES | MUL | NULL | |
| kode_kul | varchar(6) | YES | MUL | NULL | |
| nilai | int(11) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> insert into nilai values ('11010010', 'IT0101', 100);
Query OK, 1 row affected (0.03 sec)

mysql> insert into nilai values ('11010020', 'IT0101', 100);
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails ('akademik/nilai', CONSTRAINT 'nilai_ibfk_1' FOREIGN KEY ('nim') REFERENCES mhs ('nim'))
```

2. Isikan tabel seperti isian gambar di bawah :

```
mysql> select * from mhs;
+-----+-----+-----+
| nim   | nama_mhs | alamat_mhs |
+-----+-----+-----+
| 11010010 | M Rauf | Puri Kencana B2 Sleman |
| 11010011 | Rafifah Azzahra | Blunyah Gede 227 Sleman |
| 11010012 | M Tedy Farhan | Jl Boegenville IV No.9 Jambi |
| 11010013 | Shafa Dian | Jl Boegenville IV No.9 Jambi |
| 11010014 | Austin A Cetta | Palagan Regency B3 Jogja |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

```
mysql> select * from mata_kuliah;
+-----+-----+-----+-----+
| kode_kul | nama_kul | sks | sem |
+-----+-----+-----+-----+
| IT0101 | Logika Inf | 3 | 1 |
| IT0102 | Studi Islam I | 2 | 1 |
| IT0103 | Kalkulus Inf | 3 | 1 |
| IT0301 | Sertifikasi | 0 | 3 |
| IT0401 | Basis Data | 3 | 4 |
| IT0001 | Tugas Akhir | 6 | 8 |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

```
mysql> select * from nilai;
```

nim	kode_kul	nilai
11010010	IT0101	100
11010011	IT0101	100
11010010	IT0401	90
11010011	IT0401	100
11010012	IT0401	80
11010013	IT0401	85

```
6 rows in set (0.00 sec)
```

- Menampilkan hanya nama mahasiswa dan nilai untuk mata kuliah dengan kode IT0401

```
mysql> select a.nama_mhs, b.nilai from mhs a, nilai b where a.nim = b.nim and b.kode_kul = 'IT0401';
```

nama_mhs	nilai
M Rauf	90
Rafifah Azzahra	100
M Tedy Farhan	80
Shafa Dian	85

```
4 rows in set (0.00 sec)
```

- Menampilkan hanya nama mata kuliah berikut nilainya

```
mysql> select a.nama_kul, b.nilai from mata_kuliah a, nilai b where a.kode_kul = b.kode_kul;
```

nama_kul	nilai
Logika Inf	100
Logika Inf	100
Basis Data	90
Basis Data	100
Basis Data	80
Basis Data	85

```
6 rows in set (0.00 sec)
```

- Menampilkan nama mata kuliah, nama mahasiswa dan nilai yang diperoleh

```
mysql> select a.nama_kul, b.nilai, c.nama_mhs from mata_kuliah a, nilai b, mhs c where a.kode_kul = b.kode_kul and b.nim = c.nim;
```

nama_kul	nilai	nama_mhs
Logika Inf	100	M Rauf
Logika Inf	100	Rafifah Azzahra
Basis Data	90	M Rauf
Basis Data	100	Rafifah Azzahra
Basis Data	80	M Tedy Farhan
Basis Data	85	Shafa Dian

```
6 rows in set (0.00 sec)
```

6. Menampilkan nama mata kuliah, nama mahasiswa dan nilai yang diperoleh yang diurutkan dari nilai terkecil (kolom tabel disesuaikan seperti yang dihasilkan

```
mysql> select a.nama_kul as 'Nama Mata Kuliah', b.nilai as 'Nilai Mata Kuliah',  
c.nama_mhs as 'Nama Mahasiswa' from mata_kuliah a, nilai b, mhs c where a.kode_  
kul = b.kode_kul and b.nim = c.nim order by nilai asc;
```

Nama Mata Kuliah	Nilai Mata Kuliah	Nama Mahasiswa
Basis Data	80	M Tedy Farhan
Basis Data	85	Shafa Dian
Basis Data	90	M Rauf
Logika Inf	100	Rafifah Azzahra
Basis Data	100	Rafifah Azzahra
Logika Inf	100	M Rauf

6 rows in set (0.00 sec)

Relasi Tabel dengan Join

Pertemuan ke : 8

Alokasi Waktu : 2,5 Jam

Kompetensi Dasar : Praktikan dapat melakukan pencarian data antar tabel

Indikator : inner join, left join, right join, full join, union

A. Pendahuluan

Mekanisme join dipergunakan untuk mencari data dari beberapa tabel berdasarkan hubungan logis tabel-tabel tersebut. Macam-macam Join, adalah :

1. Inner Join merupakan himpunan dalam yaitu hasil gabungan dari dua buah tabel yang saling berelasi untuk semua record yang berpasangan
2. Full Outer Join mengembalikan semua baris dari kedua tabel
3. Left Outer Join menghasilkan semua baris tabel di sebelah kiri pernyataan, dan baris-baris bersesuaian dari tabel sebelah kanan pernyataan
4. Right Outer Join menghasilkan semua baris tabel di sebelah kiri pernyataan, dan baris-baris bersesuaian dari tabel sebelah kiri pernyataan
5. Union dipergunakan untuk menggabungkan dua buah operasi query ke dalam satu buah *cursor*

B. Petunjuk Praktikum

```
mysql> select * from mhs;
+-----+-----+-----+
| nim      | nama_mhs      | alamat_mhs      |
+-----+-----+-----+
| 11010010 | Muhammad Rauf | Puri Kencana B2 Sleman |
| 11010011 | Rafifah Azzahra | Blunyah Gede 227 Sleman |
| 11010012 | M Tedy Farhan  | Jl Boegenville 10 No.9 Jambi |
| 11010013 | Shafa Dian     | Jl Boegenville 10 No.9 Jambi |
| 11010014 | Austin A Cetta | Palagan Regency B3 Jogja |
| 11010015 | Luna Ramadhan  | Jl Keparakan No.1 Samarinda |
+-----+-----+-----+
6 rows in set (0.00 sec)
```

```
mysql> select * from nilai;
```

nim	kode_kul	nilai
11010010	IT0101	100
11010011	IT0101	100
11010010	IT0401	90
11010011	IT0401	100
11010012	IT0401	80
11010013	IT0401	85

```
6 rows in set (0.00 sec)
```

```
mysql> select * from mata_kuliah;
```

kode_kul	nama_kul	sks	sen
IT0101	Logika Inf	3	1
IT0102	Studi Islam I	2	1
IT0103	Kalkulus Inf	3	1
IT0301	Sertifikasi	0	3
IT0401	Basis Data	3	4
IT0801	Tugas Akhir	6	8

```
6 rows in set (0.00 sec)
```

1. Inner Join

Menampilkan nama mahasiswa, kode mata kuliah berikut nilai yang diperoleh

Dengan perintah relasi atribut antar tabel

```
mysql> select mhs.nim, mhs.nama_mhs, nilai.kode_kul, nilai.nilai
-> from mhs, nilai
-> where mhs.nim = nilai.nim;
```

nim	nama_mhs	kode_kul	nilai
11010010	Muhammad Rauf	IT0101	100
11010010	Muhammad Rauf	IT0401	90
11010011	Rafifah Azzahra	IT0101	100
11010011	Rafifah Azzahra	IT0401	100
11010012	M Tedy Farhan	IT0401	80
11010013	Shafa Dian	IT0401	85

```
6 rows in set (0.00 sec)
```

Dengan perintah inner join

```
mysql> select mhs.nim, mhs.nama_mhs, nilai.kode_kul, nilai.nilai
-> from mhs inner join nilai
-> on mhs.nim = nilai.nim;
```

nim	nama_mhs	kode_kul	nilai
11010010	Muhammad Rauf	IT0101	100
11010010	Muhammad Rauf	IT0401	90
11010011	Rafifah Azzahra	IT0101	100
11010011	Rafifah Azzahra	IT0401	100
11010012	M Tedy Farhan	IT0401	80
11010013	Shafa Dian	IT0401	85

```
6 rows in set (0.00 sec)
```

Dua gambar di atas menghasilkan hasil yang sama. Terlihat bahwa pada tabel mhs dan mata_kuliah, nim adalah atribut yang menghubungkan ke dua tabel tersebut.

Pada perintah di atas hanya record yang berpasangan yang dimunculkan. Pada tabel mhs terdapat record dengan nim '11010014' dan '11010015' yang tidak terdapat pada tabel nilai, sehingga tidak dimunculkan.

2. Left Join

Menampilkan semua data mahasiswa yang ada di tabel mhs berikut data nilai mahasiswa

```
mysql> select mhs.nim, mhs.nama_mhs, nilai.kode_kul, nilai.nilai from mhs left
join nilai on mhs.nim = nilai.nim;
```

nim	nama_mhs	kode_kul	nilai
11010010	Muhammad Rauf	IT0101	100
11010010	Muhammad Rauf	IT0401	90
11010011	Rafifah Azzahra	IT0101	100
11010011	Rafifah Azzahra	IT0401	100
11010012	M Tedy Farhan	IT0401	80
11010013	Shafa Dian	IT0401	85
11010014	Austin A Cetta	NULL	NULL
11010015	Luna Ramadhan	NULL	NULL

```
8 rows in set (0.00 sec)
```

Akan terlihat bahwa data tabel sebelah kanan akan diisi dengan NULL karena mahasiswa yang bersangkutan tidak ada nilainya, sementara data dari tabel sebelah kiri akan ditampilkan seluruhnya.

3. Right Join

Menampilkan semua data nilai yang ada di tabel nilai berikut nama mahasiswa.

```
mysql> select mhs.nim, mhs.nama_mhs, nilai.kode_kul, nilai.nilai from mhs right
join nilai on mhs.nim = nilai.nim;
```

nim	nama_mhs	kode_kul	nilai
11010010	Muhammad Rauf	IT0101	100
11010011	Rafifah Azzahra	IT0101	100
11010010	Muhammad Rauf	IT0401	90
11010011	Rafifah Azzahra	IT0401	100
11010012	M Tedy Farhan	IT0401	80
11010013	Shafa Dian	IT0401	85

```
6 rows in set (0.00 sec)
```

Akan terlihat bahwa data dari tabel sebelah kanan yaitu tabel nilai akan ditampilkan seluruhnya.

4. Union

```
mysql> select * from mhs union select * from nilai;
```

nim	nama_mhs	alamat_mhs
11010010	Muhammad Rauf	Puri Kencana B2 Sleman
11010011	Rafifah Azzahra	Blunyah Gede 227 Sleman
11010012	M Tedy Farhan	Jl Boegenville IU No.9 Jambi
11010013	Shafa Dian	Jl Boegenville IU No.9 Jambi
11010014	Austin A Cetta	Palagan Regency B3 Jogja
11010015	Luna Ramadhan	Jl Keparakan No.1 Samarinda
11010010	IT0101	100
11010011	IT0101	100
11010010	IT0401	90
11010011	IT0401	100
11010012	IT0401	80
11010013	IT0401	85

12 rows in set (0.02 sec)

Subquery

Pertemuan ke : 9

Alokasi Waktu : 2,5 Jam

Kompetensi Dasar :

1. Praktikan memahami konsep SubQuery
2. Praktikan dapat membuat perintah SubQuery

Indikator : IN, NOT IN, EXIST, Operator Perbandingan

A. Pendahuluan

Sub Query adalah suatu query yang menjadi bagian dari suatu query, digunakan untuk menangani masalah yang kompleks yang mungkin sulit untuk dilakukan hanya dengan sebuah query. Sub Query menyediakan cara alternatif untuk melakukan operasi yang membutuhkan join atau union yang rumit. Beberapa aturan :

1. Dalam sebuah query boleh memiliki lebih dari 1 sub query.
2. Sebuah sub query boleh memiliki sub query lagi.
3. Operator perbandingan yang dapat digunakan adalah =, >, <, >=, <=, <>, !=, <=>, IN, ANY, SOME, ALL, EXISTS, NOT EXISTS

Untuk operator =, >, <, >=, <=, <>, !=, <=> sub querynya hanya boleh memiliki 1 baris, jika barisnya memiliki lebih dari 1 baris akan menampilkan pesan “Subquery returns more than 1 row”.

Untuk operator IN akan memeriksa apakah suatu nilai di outer query ada dalam sebuah hasil sub query. Sub query boleh memiliki data lebih dari 1 baris. Operator IN bisa disamakan dengan operator “= ANY”. Lawan hasil dari operasi IN adalah NOT IN. Operator NOT IN bisa disamakan dengan “<> ALL”

Untuk operator EXISTS digunakan untuk memeriksa apakah subquery memiliki baris atau tidak. Jika minimal ada 1 baris (walaupun hanya berisi NULL), maka akan bernilai TRUE. NOT EXISTS adalah kebalikan dari EXISTS.

B. Petunjuk Praktikum

Tabel Mhs (NIM, Nama_mhs, Alamat_mhs)

Tabel Nilai (NIM, Nilai, Kode_kul)

Tabel Kuliah (NIM, Kode_kul)

Isikan tabel nilai seperti di bawah :

```
mysql> select * from nilai;
+----+-----+-----+
| nim      | kode_kul | nilai |
+----+-----+-----+
| 11010010 | IT0101   | 80    |
| 11010011 | IT0101   | 75    |
| 11010010 | IT0401   | 70    |
| 11010011 | IT0401   | 75    |
| 11010012 | IT0401   | 70    |
| 11010013 | IT0401   | 70    |
+----+-----+-----+
6 rows in set (0.00 sec)
```

1. Menampilkan data nilai mahasiswa yang nilainya melebihi rata-rata nilai mata kuliah secara keseluruhan

Query membutuhkan Sub Query karena untuk dapat mencari data yang diinginkan, maka rata-rata mata kuliah harus dicari terlebih dahulu.

Sub query bekerja untuk mencari nilai rata dari matakuliah yang akan menjadi patokan dalam seleksi data nilai

Sub Query : `SELECT avg(nilai) from nilai`

```
mysql> select avg(nilai) from nilai;
+-----+
| avg(nilai) |
+-----+
| 73.3333    |
+-----+
1 row in set (0.05 sec)
```

Main Query : `SELECT nim, nilai FROM nilai WHERE nilai > (SELECT avg(nilai) from nilai)`

```
mysql> select nim, nilai from nilai where nilai > (select avg(nilai) from nilai);
+----+-----+
| nim      | nilai |
+----+-----+
| 11010010 | 80    |
| 11010011 | 75    |
| 11010011 | 75    |
+----+-----+
3 rows in set (0.03 sec)
```

2. Carilah data nilai yang nilainya sama dengan nilai terbesar.

Query membutuhkan Sub Query karena untuk dapat mencari data nilai yang diinginkan, maka nilai terbesar harus dicari terlebih dahulu.

Query ini bisa dilakukan dengan menggunakan ORDER BY dan LIMIT, tetapi hanya akan menghasilkan 1 baris saja. Bagaimana jika dagolтта yang sesuai dengan kriteria lebih dari 1 baris.

```
mysql> select * from nilai;
+----+-----+-----+
| nim      | kode_kul | nilai |
+----+-----+-----+
| 11010010 | IT0101   | 80    |
| 11010011 | IT0101   | 75    |
| 11010010 | IT0401   | 70    |
| 11010011 | IT0401   | 75    |
| 11010012 | IT0401   | 70    |
| 11010013 | IT0401   | 80    |
+----+-----+-----+
6 rows in set (0.00 sec)
```

```
SELECT nim, nilai, kode_kul
FROM nilai WHERE nilai=(SELECT MAX(nilai)
                        FROM nilai)
```

```
mysql> select nim, nilai, kode_kul from nilai where nilai = (select max(nilai) f
rom nilai);
+----+-----+-----+
| nim      | nilai | kode_kul |
+----+-----+-----+
| 11010010 | 80    | IT0101   |
| 11010013 | 80    | IT0401   |
+----+-----+-----+
2 rows in set (0.00 sec)
```

```
SELECT nim, nilai, kode_kul
FROM nilai ORDER BY nilai asc LIMIT 1
```

Query pertama mungkin menampilkan data nilai lebih dari 1 baris ketika baris yang nilainya sama dengan nilai MAX(nilai) lebih dari 1 baris.

Query kedua hanya akan menampilkan 1 baris saja karena ada penggunaan LIMIT. Kekurangan dari SQL ini adalah ketika ada data nilai yang sama-sama memiliki nilai sama dengan MAX(nilai) lebih dari 1 baris.

```
mysql> select * from nilai order by nilai desc limit 1;
+-----+-----+-----+
| nim      | kode_kul | nilai |
+-----+-----+-----+
| 11010010 | IT0101   | 80    |
+-----+-----+-----+
1 row in set (0.00 sec)
```

3. Menampilkan data mahasiswa yang mengambil mata kuliah 'IT0401'

Query tersebut membutuhkan sub query karena harus melakukan perbandingan data mahasiswa dengan data mahasiswa yang ada di tabel kuliah. Berarti data mahasiswa yang ada di tabel kuliah harus dicari terlebih dahulu.

```
SELECT nim, nama_mhs
FROM mhs
WHERE nim IN (SELECT nim FROM kuliah
              where kode_kul = 'IT100')
```

Sub query bekerja untuk mencari data mhs yang telah terdaftar di tabel kuliah (mengambil mata kuliah tsb)

```
mysql> select nim, nama_mhs from mhs where nim in (select nim from nilai where
kode_kul = 'IT0401');
+-----+-----+
| nim      | nama_mhs |
+-----+-----+
| 11010010 | Muhammad Rauf |
| 11010011 | Rafifah Azzahra |
| 11010012 | M Tedy Farhan |
| 11010013 | Shafa Dian |
+-----+-----+
4 rows in set (0.00 sec)
```

4. Menampilkan data mahasiswa yang tidak mengambil mata kuliah basis data

```
mysql> select nim, nama_mhs from mhs where nim not in (select nim from nilai wh
ere kode_kul = 'IT0401');
+-----+-----+
| nim      | nama_mhs |
+-----+-----+
| 11010014 | Austin A Cetta |
| 11010015 | Luna Ramadhan |
+-----+-----+
2 rows in set (0.00 sec)
```